Data General

User's Manual

microNOVA™ DTOS

015-000059-03

NOTICE

Data General Corporation (DGC) has prepared this manual for use by DGC personnel, licensees, and customers. The information contained herein is the property of DGC and shall not be reproduced in whole or in part without DGC's prior written approval.

Users are cautioned that DGC reserves the right to make changes without notice in the specifications and materials contained herein and shall not be responsible for any damages (including consequential) caused by reliance on the materials presented, including, but not limited to typographical, arithmetic, or listing errors.

NOVA, **ECLIPSE** and **INFOS** are registered trademarks of Data General Corporation, Westboro, Massachusetts. **DASHER** and **microNOVA** are trademarks of Data General Corporation, Westboro, Massachusetts.

Ordering No. 014-000059
© Data General Corporation 1976,1977,1978
All Rights Reserved
Printed in the United States of America
Rev. 03, July, 1978

CONTENTS

CHAPTER I	**INTRODUCTION TO microNOVA™ DTOS**
I-1	OPERATION SYSTEM
I-1	Device Testing
I-1	Equipment Table
I-1	RELATED DOCUMENTS
CHAPTER II	**FUNCTIONAL DESCRIPTION**
II-1	INTERNAL STRUCTURE
II-1	Monitor
II-1	Mini-Monitor
II-1	Parameter Storage
II-1	Equipment Table
II-1	DIAGNOSTIC TESTS
II-1	Composition
II-2	Simple Diagnostics
II-2	Complex Diagnostics
II-2	Overnight Testing
II-2	Error Detection
II-2	CONTROL TRANSFERS
II-2	Program Loading
II-2	Debuggers
II-2	Monitor Reloading
II-2	PROGRAM MODES
II-2	Auto Mode
II-3	Semi-auto Mode
II-3	Manual Mode
II-4	Debug Mode
II-4	Data Channel Test Mode

CONTENTS (Continued)

CHAPTER III	**SYSTEM COMMANDS**
III-1	INTRODUCTION
III-1	Operator Prompt
III-1	Rubout Key
III-1	Mnemonics
III-1	COMMANDS WHICH RUN ALL APPLICABLE DIAGNOSTICS
III-2	RUN
III-2	ACCEPT
III-2	RUNALL
III-2	COMMANDS WHICH RUN ONLY SPECIFIED DIAGNOSTICS
III-2	Command Format
III-2	DEBUG
III-3	ODT
III-3	SELECT
III-3	REPEAT
III-3	LOAD
III-3	LOAD.n
III-3	COMMANDS WHICH RUN THE DATA CHANNEL TEST
III-4	CRUN
III-4	CRUNALL
III-4	CSELECT
III-4	CREPEAT
III-4	CLOAD
III-4	CLOAD.n
III-5	COMMANDS WHICH MODIFY THE EQUIPMENT TABLE
III-5	ADD
III-5	DELETE
III-5	ASSIGN
III-5	MISCELLANEOUS COMMANDS
III-5	EQUIP
III-5	HELP
III-5	DIR
III-5	ENTER
III-5	QUIT
III-5	SWREG

CONTENTS (Continued)

APPENDIX A	BOOTSTRAP PROCEDURES
APPENDIX B	PROBLEM ISOLATION
APPENDIX C	COMMAND SUMMARY
APPENDIX D	EQUIPMENT TABLE
APPENDIX E	ERROR MESSAGES
APPENDIX F	USE OF THE SWITCH REGISTERS IN DIAGNOSTIC TESTING
APPENDIX G	DIAGNOSTIC ABSTRACTS
APPENDIX H	SYSTEM PROGRAMS
APPENDIX I	DIAGNOSTIC OPERATING PROCEDURE

This page intentionally left blank.

CHAPTER I
INTRODUCTION TO microNOVA™ DTOS

OPERATING SYSTEM

MicroNOVA DTOS is a flexible, compact and comprehensive diagnostic operating system for microNOVA processors and peripheral equipment. The entire system, including supervisory programs and diagnostic tests, is contained on a single diskette, easy to carry and store.

Device Testing

The operator controls device testing with online commands issued to the processor through a terminal keyboard. These commands allow the operator to load and run any diagnostic contained on the diskette and to select the I/O device to be tested. Certain commands also allow the operator to run a data channel test concurrently with a diagnostic. Commands allow for continuous overnight running of a loop containing all diagnostics applicable to a particular installation. A hard copy, available in the morning, indicates how often each test failed and where the failure occured.

Two debuggers provide for breakpoints that allow the user to isolate program errors, start execution at any location, and examine and alter the contents of memory locations, accumulators, stack and frame pointers.

Equipment Table

MicroNOVA DTOS constructs an equipment table in memory containing the mnemonic and device code for each piece of equipment on the processor data bus it is capable of identifying. Devices contained in this table will be tested automatically when certain commands are issued. Devices not included in the table may be added to it with commands.

RELATED DOCUMENTS

In addition to this manual, two other DGC publications may be helpful when using microNOVA DTOS. They are:

Programmer's Reference, microNOVA Computers (DGC No. 015-000050)

Technical Reference, microNOVA Computer Systems (DGC No. 014-000073)

This page intentionally left blank.

CHAPTER II
FUNCTIONAL DESCRIPTION

INTERNAL STRUCTURE

MicroNOVA DTOS is an operating system which consists of a collection of diagnostic tests and a supervisory program contained on a diskette. To bring microNOVA DTOS into memory, the user must follow one of the bootstrap procedures described in Appendix A. When one of these bootstrap procedures has been initiated, the monitor is loaded into memory.

Monitor

The monitor handles all communication between the operator and microNOVA DTOS. This communication is through commands typed on a terminal by the operator. The monitor is ultimately responsible for loading all diagnostic tests and providing feedback from microNOVA DTOS to the operator. The monitor begins each operation by loading a small program, called the mini-monitor, into the top 200_8 locations of writeable memory. The monitor loads all of each diagnostic into memory except that portion which would occupy locations $0\text{-}377_8$.

Mini-monitor

The function of the mini-monitor is to load into memory locations $0\text{-}377_8$ any diagnostic that the user specifies to be run. In addition, the mini-monitor is responsible for reloading and starting the monitor after a diagnostic relinquishes control.

Parameter Storage

As diagnostics are scheduled to be run, the monitor records parameter information about each diagnostic in the area of memory occupied by the mini-monitor. These parameters contain such information as the starting address, the size, and the device codes to be used for each diagnostic.

The memory checking diagnostics use the area of memory in which the mini-monitor and the parameters reside. Before one of these diagnostics write in this area, it relocates this information into another area of memory. After the completion of the test, the diagnostic will write the mini-monitor and the parameters into their original locations in memory.

Equipment Table

As each diagnostic is scheduled to run, the monitor searches the I/O bus to determine which devices are to be recognized by the diagnostics as test devices. The monitor constructs a list of these devices in memory, called the equipment table. The user can add devices to the equipment table or remove them by means of commands.

DIAGNOSTIC TESTS

Throughout this manual the term, *diagnostic,* is used to refer to both diagnostic and reliability test programs. Although both types of programs are used to check the working condition of devices, there is a difference between them. A reliability test measures the ability of a device to function without a failure. When a failure is indicated, the test does not point out its exact nature. A diagnostic test will discover a failure or potential failure and also pinpoint its location.

Composition

Most DGC diagnostics consist of a series of simple tests. Each one sends a particular combination of input signals to a small portion of a specified unit and performs some simple test on the output. Generally, each test is initialized by a small subroutine. This subroutine sets the internal pass counters (that is, the number of passes to be made through the test), establishes the proper address for the diagnostic to jump to after each pass, and determines any other parameters necessary to run the test. Another small subroutine keeps track of the number of times the diagnostic has been run successfully. This subroutine is responsible for having the diagnostic jump to a particular address after it has been run the number of times required. If the diagnostic has not completed the established number of runs, the subroutine forces the diagnostic to begin again.

Simple Diagnostics

On some of the simple diagnostics, a failing test will cause the processor to halt. At this time, one of the following three operations can occur: 1) If the hardware ODT is present, the ODT routine will be executed. It will print the program counter (PC) location of the halt, plus one. Use ODT commands, at this time, to examine all the accumulators and carry. 2) If the Hand Held Console is present, the HHC routine will be executed. It will display the program counter (PC) location of the halt, plus one, and wait for operator commands. 3) If none of the above is present, the CPU will halt and the RUN light on the main chassis front panel will be turned off. The user must consult the listing of the diagnostic test that was running to find the exact reason for the failure. A failure in this type of diagnostic almost precludes passing any other diagnostic.

Complex Diagnostics

On the more complex diagnostics, a subroutine will force the diagnostic to print the memory location where the failure occurred and return control to the monitor program when a command has been issued which causes all applicable diagnostics to run in sequence. The monitor will print the name of the diagnostic in which the failure occurred on each run except the first.

Overnight Testing

DTOS was designed to return control to the monitor if an error occurs in a complex diagnostic when running in auto or semi-auto mode. Thus, the system will not *hang up* on a single failure when an overnight run is attempted. When control returns to the monitor, the next scheduled test, if any, will be loaded and run. Consequently, to determine whether any errors have occurred, the user must examine the terminal output.

Error Detection

If any errors have occurred in overnight testing, the diagnostic in which they occurred should be loaded and run individually. The **LOAD** command is used for this operation. (See Chapter III, section entitled *Commands Which Run Only Specified Diagnostics.)* The **SWREG** command can be used before loading the program (with a **LOAD, DEBUG** or **ODT** command). This allows the operator to use the terminal keyboard to either print the failure rate in percent, inhibit the printing of error messages, or force the diagnostic to continue with the next internal pass after a failure occurs. (See Appendix F for keyboard and console data switches.)

CONTROL TRANSFERS

In order for memory to accommodate both the monitor program and a diagnostic, control transfers must take place so that each of these programs is loaded into memory at the appropriate time. Since diagnostics are scheduled to load and run by user commands, the monitor program stores parameter information concerning the diagnostics in an area of memory. These parameters are referred to by the monitor each time a diagnostic is to be loaded into memory.

Program Loading

The monitor program loads all of the diagnostic program, except for locations 0-377_8 , which are loaded by the mini-monitor program. After the diagnostic has been loaded into memory, it takes control of the system and begins testing the appropriate devices.

Debuggers

An exception to this is made when one of the two debugger programs is scheduled to be loaded and run with a diagnostic. Control then passes from the monitor to the debugger and the diagnostic may be started with debugger commands.

Monitor Reloading

When the scheduled diagnostics have finished running, control of the system passes to the mini-monitor program, which automatically reloads the monitor by writing over the diagnostic program in residence.

PROGRAM MODES

Diagnostics are executed under microNOVA DTOS in one of five modes: auto, semi-auto, manual, debug, or data channel test.

Auto Mode

In auto mode microNOVA DTOS compares its equipment table with the equipment requirements of the test programs as shown in the directory; then, it sequentially executes those programs that exercise the devices on the machine under test. Each test program is loaded and executed automatically. At its conclusion the test program returns control to microNOVA DTOS so that the next program can be run. No operator communication is required after the initial command. The commands that run diagnostics in auto mode are **RUN**, **ACCEPT**, and **RUNALL**. (See Chapter III.)

Semi-auto Mode

If communication with the operator is required, the program may be run in semi-auto mode. (It cannot be run in auto mode.) In all other respects operation in the two modes is similar. One or more test programs may be specified in the initial command to microNOVA DTOS and these programs will be executed sequentially with return being made to microNOVA DTOS after each program ends. The commands that run diagnostics in semi-auto mode are **LOAD.n**, **SELECT** and **REPEAT**. (See Chapter III.)

Manual Mode

Manual mode is used when a return to microNOVA DTOS is not desired. With the initial command, DTOS loads and starts the program. At the conclusion of the test, the program loops back to the start of the test. This mode is useful when the operator is troubleshooting a machine that is failing a particular test. The **LOAD** and **CLOAD** commands cause DTOS to operate in manual mode.

Debug Mode

If the **ODT** command has been given, the **ODT** debugger is loaded just below the mini-monitor, and its entry address is printed on the output device. If the **DEBUG** command has been given, the debugger is loaded at location $12,000_8$, then the test program is loaded. The microNOVA DTOS transfers control to the debugger rather than to the starting address of the test program. This procedure is useful for things other than debugging. For example, a program can be loaded with **DEBUG** or **ODT** and the program will not run until the operator starts it.

Once the test program is started from the debugger, operation is identical to manual mode. The test program will loop on completion without returning to microNOVA DTOS. It should be noted that the test program is free to write over the debugger, and if this occurs, the debugger becomes useless after the program starts. Individual diagnostic listings indicate whether or not the debugger will be written over.

Data Channel Test Mode

MicroNOVA DTOS provides two data channel reliability test programs, MNCAT and MKITTEN, which may be multi-programmed with diagnostics. These tests are loaded with either the **CRUN**, **CRUNALL**, **CSELECT**, **CREPEAT**, **CLOAD**, or **CLOAD.n** commands. When one of these commands is issued, the monitor searches the I/O bus for the presence of a DGC field service I/O test board. If found, it is used as a test device for the MKITTEN program. If not found, the diskette will be used as a test device for the MNCAT program. After the data channel test has been selected by the monitor, it is loaded along with the scheduled diagnostic. Command execution then begins with the diagnostic, making one pass alone. Subsequent passes are made with the diagnostic running in the foreground and the data channel test running in the background. Thus, those commands which run a diagnostic through only one pass do not respond to the C option.

NOTE *Foreground and background programming is explained in detail in the Real Time Disc Operating System (RDOS) Reference Manual (DGC No. 093-000075).*

Once started, the data channel test begins reading and writing data patterns to the I/O test board or the diskette. The MNCAT or MKITTEN programs are interrupt driven and are totally transparent to the running diagnostic programs. Upon receiving an interrupt from the data channel test device, control is transferred to the data channel test program; after performing its operation, that program transfers control back to the diagnostic. Each time the test program is started, an **S** is printed on the output device.

The data patterns used to test the data channel bus are all ones, all zeros and an increment/swap pattern. Once all patterns have been run, a **P** is printed on the output device and the patterns are started again from the beginning.

Stand-alone versions of MSCAT and MSKITTEN, also contained on the microNOVA diskette, allow the user to run the data channel test with paper tape diagnostics or with their own programs.

NOTES *User-written programs should refrain from using the IORST instruction because MSCAT and MSKITTEN are interrupt driven.*

MSCAT and MSKITTEN will replace location 1 in a user-written program with a pointer to their own internal interrupt handler and will store the contents of the user's location 1 for use in vectoring interrupts, other than the data channel test, back to the user's program. If location 1 is modified after the data channel test has started, the user should vector all interrupts from the diskette or I/O test board back to the data channel test program.

Stand-alone MSCAT and MSKITTEN must be loaded before the diagnostic program. When either program is loaded, the data channel test program starts automatically. It sizes memory, writes itself into the top of memory, and sends a message to the output device, giving the value of the top location in memory and telling the operator to load the diagnostic program. After the diagnostic has been loaded, the data channel test can be started in one of two ways:

1) by inserting the following at the beginning of the diagnostic program:

```
MEMTOP (-1377 OR -1375) OCTAL
JSR  .-1
```

2) by restoring location 0 of the diagnostic from the listing, then starting the program at Memtop -377_8.

In either case, clear the SWREG location (Memtop -122_8) to 0, and place the test unit's device code in location Memtop -121_8.

NOTE *The MSCAT and MSKITTEN programs may be started in either location MEMTOP -1377_8 or location MEMTOP -1375_8. Starting the test at location MEMTOP -1377_8, will mask out TTO AND TTI interrupts. Starting the test at location MEMTOP -1375_8 will mask out only TTO interrupts.*

This page intentionally left blank.

CHAPTER III
SYSTEM COMMANDS

INTRODUCTION

Communication between the user and microNOVA DTOS is through commands typed on a terminal keyboard. Commands may be used to load and run diagnostics, to modify the equipment table, to print reference information on hard copy output, and to reload the binary and bootstrap loaders after testing is over.

Operator Prompt

An asterisk printed on the terminal following a successful load of the monitor program prompts the operator to issue commands. The commands must follow the format outlined in this manual and should be typed directly after the asterisk and followed by a carriage return.

Rubout Key

The RUBOUT key is used to delete erroneously typed characters from a microNOVA DTOS command string. Each time the RUBOUT key is depressed, the right-most character is deleted. When a character is deleted, it is echoed on the terminal.

Mnemonics

Commands whose format requires that a diagnostic name or a device name be supplied by the operator work only when the standard mnemonic for the diagnostic or device is typed in. See Appendix G for diagnostic mnemonics and Appendix D for device mnemonics.

COMMANDS WHICH RUN ALL APPLICABLE DIAGNOSTICS

RUN, **RUNALL**, and **ACCEPT** are the three commands which can be used to sequentially execute all diagnostics applicable to the devices contained in the equipment table. When one of these commands is issued, the device codes and mnemonics of all devices attached to the I/O bus that are recognized by microNOVA DTOS will be printed on the terminal.

After a **RUN** or **ACCEPT** command has been issued, the monitor will print the mnemonic and revision number of each diagnostic as it is loaded. When all applicable diagnostics have been run, the monitor will be reloaded and an asterisk will be printed on the terminal, prompting the operator for additional commands. After a **RUNALL** command has been issued, the monitor will print the mnemonic and revision number of each diagnostic as it is loaded the first time. This command causes all applicable diagnostics to be loaded and executed sequentially until an error occurs in a simple diagnostic or until the processor is stopped manually.

Each time the group of all applicable diagnostics has been run, a number reflecting the number of times the group has been run will be printed on the terminal.

RUN

This command causes all diagnostics that apply to devices in the current equipment table to load and run sequentially. The internal pass counters will be set by the diagnostics. After all the diagnostics have been run, the monitor is reloaded and an asterisk is printed on the terminal prompting the user to issue additional commands.

ACCEPT

This command causes all diagnostics applicable to the current equipment table to load and run sequentially. Internal pass counters will be set to one. After all diagnostics have made one pass, the monitor is reloaded and an asterisk is printed on the terminal, prompting the user to issue additional commands.

NOTES *The only difference between these two commands is that the ACCEPT command makes only one pass per test within a diagnostic and the RUN command makes as many passes per test within a diagnostic as the diagnostic specifies. Consequently, the ACCEPT command has a much shorter execution time than the RUN command.*

If an error should occur while a diagnostic is running after either of these commands has been issued, the contents of the program counter at the time the error occurs will be printed on the terminal, control will return to the monitor and the name of the test in which the error occurred will be printed.

RUNALL

This command causes all diagnostics applicable to devices in the current equipment table to load and run sequentially. The internal pass counters will be set by the diagnostics. After all diagnostics have run, the message **END RUN** *x* will be printed on the terminal, *x* signifying the number of times all applicable diagnostics have been run. All applicable diagnostics will load and run repeatedly until the machine is stopped manually.

If an error occurs during the execution of a diagnostic in the first run of all applicable tests, the contents of the program counter at the time the error occurs will be printed on the terminal. If an error occurs in a diagnostic in any run after the first of all applicable tests, the contents of the program counter at the time the error occurs will be printed on the terminal followed by
ERROR FOUND IN *name of diagnostic.*
In both cases, control returns to the monitor.

COMMANDS WHICH RUN ONLY SPECIFIED DIAGNOSTICS

The **DEBUG**, **ODT**, **SELECT**, **REPEAT**, **LOAD**, and **LOAD.n** commands cannot be executed until the operator specifies a diagnostic name or names. The diagnostic must be contained in the program directory of the diskette used; otherwise an error message will be printed on the terminal.

Command Format

The user specifies a diagnostic by typing one of the above commands followed by a space and the mnemonic of the diagnostic exactly as it appears in the program directory. Each mnemonic consists of 9 characters including spaces. The name format requires that all spaces be typed, except those at the end of a name. The mnemonic should be followed by a carriage return.

NOTE *None of the diagnostics which test peripheral devices will run in AUTO mode.*

DEBUG *diagnostic mnemonic*

This command causes the Symbolic Debugger program to be loaded with the specified diagnostic. After both programs have been loaded, the starting address of the diagnostic and the starting address of the debugger will be printed on the terminal to enable the user to begin execution of the diagnostic with commands from the debugger. The debugger is then started.

Locations 50_8 through 57_8 are the areas of memory reserved for breakpoints by the debugger programs.

ODT *diagnostic mnemonic*

This command (used with processors having the hand-held console) causes the debugger program, Octal Debugging Tool, to be loaded with the specified diagnostic program. After the diagnostic has been loaded, it may be started with the commands from the ODT program. Refer to Programmers Reference, microNOVA Computers *(DGC no. 015-000050)* for instructions for setting breakpoints.

> **NOTE** *A hardware version of the ODT program also exists. It is contained on a ROM chip inside the microNOVA processors not having the hand-held console. It occupies the top locations of memory. The hardware and software versions of the ODT programs may be identified by their prompts. The prompt returned by the software program is an ampersand - &. The prompt returned by the hardware program is an exclamation point - !. The software ODT does not respond to the "L" command (PROG. LOAD).*

SELECT *diag. mnem. diag. mnem.*

This command will cause the specified diagnostic tests to load and execute in sequence. Internal pass counters will be set by the diagnostics. After all diagnostics have been executed, the monitor will be reloaded and an asterisk will be printed on the terminal prompting the user to issue additional commands.

Example:

The command **SELECT MNLGCT MOSMEM** will cause the LOGIC TEST and the **MOS MEMORY TEST** to load and execute in sequence.

REPEAT *diag. mnem. diag. mnem.*

This command causes the specified diagnostics to be loaded and executed repeatedly until the processor is stopped manually.

Example:

The command **REPEAT MNLGCT MOSMEM** will cause the LOGIC TEST and the MOS MEMORY TEST to load and execute repeatedly until the processor is stopped manually.

LOAD *diagnostic mnemonic*

This command causes the specified diagnostic to load and execute repeatedly until either an error is found by the diagnostic or until the processor is stopped manually. Internal pass counters are set by the diagnostic. After the machine has halted, microNOVA DTOS must be reloaded by following the instructions for the restart procedure described in Appendix A.

Example:

The command **LOAD ARITHTST** will cause the diagnostic, ARITHTST, to be loaded and executed. The diagnostic will run until stopped manually or until an error occurs.

LOAD.n *diagnostic mnemonic*

This command causes the specified diagnostic to load and run a specified number of times; *n* signifies the number of times the diagnostic is to be executed and must be stated in octal notation. Internal pass counters will be set by the diagnostic. After the diagnostic has been loaded and executed the specified number of times, the monitor will be reloaded and an asterisk will be printed on the terminal, prompting the operator for additional commands.

Example:

The command **LOAD.25 MNSCMT** will cause the MNSCMT diagnostic to be loaded and run 25_8 times.

COMMANDS WHICH RUN THE DATA CHANNEL TEST

CRUN, CRUNALL, CSELECT, CREPEAT, CLOAD, and **CLOAD.n** are the commands used to multi-program the data channel reliability test concurrently with diagnostics. See the end of Chapter II for a complete description of the data channel test.

A description of each of these commands follows:

CRUN

This command causes all diagnostics in the program directory which are applicable to the processor and the I/O devices to be loaded and executed concurrently with the data channel test program. Each diagnostic will be executed first by itself, and then make subsequent passes with the data channel test program. After all applicable diagnostics have been run, the monitor will be reloaded and an asterisk will be printed on the terminal prompting the user to issue additional commands.

CRUNALL

This command causes all diagnostics in the program directory which are applicable to the processor and the I/O devices to be run concurrently with the data channel test program. Each diagnostic will be run first by itself, and then make subsequent passes with the data channel test program. After all applicable diagnostics have been run, the message, **END RUN** *x* , will be printed on the terminal, *x* signifying the number of times that the entire list of diagnostics has been run. The diagnostics will run repeatedly until the processor is stopped manually.

If an error occurs during the execution of a complex diagnostic after either of these commands has been issued, the contents of the program counter at the time the error occurs will be printed on the terminal. For the **CRUNALL** command the contents of the program counter at the time the error occurs will be printed on the terminal, and, except during the first run will be followed by:

ERROR FOUND IN *name of diagnostic.*

In all cases, if an error occurs during the running of a diagnostic, control returns to the monitor.

CSELECT *diag. mnem. diag. mnem.*

This command causes the specified diagnostics to be loaded and executed in sequence concurrently with the data channel test program. Each diagnostic will make one pass alone, and then makes subsequent passes with the microNOVA CAT or microNOVA KITTEN program running in the background. When all the specified tests have been run, the monitor will be reloaded and an asterisk will be printed on the terminal, prompting the user to issue additional commands.

If an error occurs in the execution of a simple diagnostic, the machine halts and microNOVA DTOS must be reloaded with one of the bootstrap procedures described in Appendix A.

CREPEAT *diag. mnem. diag. mnem.*

This command causes the specified dianostics to be loaded and executed concurrently with the data channel test program in a repetitive sequence. Each diagnostic will make one pass alone, and then make subsequent passes with the microNOVA CAT or microNOVA KITTEN program running in the background. This process will take place until the processor is stopped manually or until an error is found by a diagnostic. After the machine has stopped, microNOVA DTOS must be reloaded with one of the bootstrap procedures described in Appendix A.

CLOAD *diagnostic mnemonic*

This command causes the specified diagnostic to be loaded and executed concurrently with the data channel test program. The diagnostic will make one pass alone, and then make subsequent passes with the diagnostic running in the foreground and the data channel test running in the background. This process will take place repeatedly until the processor is stopped manually or until an error is found by a diagnostic. After the machine has stopped, microNOVA DTOS must be reloaded with one of the bootstrap procedures described in Appendix A.

CLOAD.n *diagnostic mnemonic*

This command causes the specified diagnostic to be loaded and run concurrently with the data channel test program a specified number of times; *n* signifies the number of times the diagnostic is to run and must be stated in octal notation. The first pass of the diagnostic will run alone; then subsequent passes are made with the diagnostic running in the foreground and the data channel test running in the background. After the diagnostic has been run the specified number of times, the monitor will be reloaded and an asterisk will be printed on the terminal, prompting the user to issue additional commands.

Example:

The command **CLOAD.25 ARITHTST** will cause the ARITHMETIC TEST diagnostic to be loaded and executed concurrently with the data channel test 24_8 times after the first pass is run alone.

COMMANDS WHICH MODIFY THE EQUIPMENT TABLE

The **ADD**, **DELETE**, and **ASSIGN** commands allow the operator to modify the current equipment table and thereby control which devices will be tested. Using these commands, the operator may add devices to the table, remove devices from the table, or replace the current table with mnemonics for another group of devices. The command format requires the user to specify both the standard device code and mnemonic for each device that will be added to or removed from the table.

After the equipment table has been modified with one of these commands, it will remain in a modified state until the next time that a microNOVA DTOS bootstrap procedure is performed.

The following is a description of each command which can be used to modify the equipment table:

ADD *device name, device code*

This command causes the specified devices to be added to the equipment table.

DELETE *device name, device code*

This command causes the specified devices to be removed from the equipment table.

ASSIGN *device name, device code*

This command causes all devices in the equipment table to be deleted and the specified devices to be added, thereby replacing the current equipment table with the specified devices.

MISCELLANEOUS COMMANDS

The following commands perform a variety of functions which aid in using microNOVA DTOS.

EQUIP

This command causes the mnemonics and device codes in the current equipment table to be printed on the terminal.

HELP

This command causes the command summary as shown in Appendix C and a list of all devices that may be tested automatically if attached to the I/O bus to be printed on the terminal.

DIR

This command causes the mnemonics of all diagnostics in the program directory to be printed on the terminal.

ENTER *heading information*

This command allows the user to put headings on the printed teletype or punched paper tape record (if the terminal is an ASR). Input is terminated by the control Z character. A line feed or carriage return is echoed by a carriage return, line feed sequence. The input data is not stored in memory.
Example:

```
ENTER MARCH 1977
TEST 1
```

This command will cause the following to be printed on the terminal and the punched paper tape (ASR terminals only):

```
MARCH 1977
TEST 1
```

QUIT

This command loads and starts the exerciser program (MNMRT).

SWREG

This command displays the current value of the switch register and allows the user to change this value.

The switch register is a 16-bit register in which a bit is a switch that alters a program function. Appendix F lists and describes all of these switches.

After the user inputs the SWREG command, the terminal prompts with the following:

INPUT IN OCTAL AND CR

SWREG = XXXXXX

The "XXXXXX" is a 6-digit octal number which represents the switch settings in the register. This number expands to a 16-bit number which corresponds to the 16 switches in the register. Incidentally, the left-most bit (bit 0) of this number is switch 0.

The user turns these switches "ON" or "OFF" by inputting a 6-digit octal number which is coded like "XXXXXX". Also, the user presses the carriage return to exit this command after changing the switch register.

Since the user may input this command to only display the current value of the switch register, he presses the line feed key to exit this command without changing the value of the switch register. (Entering a carriage return here would make all the switch values zero by default.)

Example:

SWREG

INPUT IN OCTAL AND CR

SWREG = XXXXXX 002000 (carriage return)

The "002000" input in the above example enables the print-out to the line printer because bit 5 is a 1 (switch 5 is "ON"). All the other switches are turned off. The two leading zeroes did not have to be included in the input because DTOS zero-fills any leading positions that are left blank.

APPENDIX A
BOOTSTRAP PROCEDURES

INTRODUCTION

MicroNOVA DTOS initially is brought into memory with the first program on the diskette, the bootstrap loader. The procedure for initiating the bootstrap loader varies, depending upon the presence of various options on the microNOVA processor. Bootstrap procedures are described below.

MicroNOVA DTOS DISKETTE should be present in disk drive 0 (selected by jumpers) with the **READY** light illuminated on the diskette front panel.

If the microNOVA ODT ROM is present, enter the device code for the diskette on the terminal followed by the letter, L.

If the hand-held console is present, press the **RESET** key; press the **CLR D** key; enter the device code for the diskette on the console; then press the **PROGRAM LOAD** key.

If the processor program load option is present and jumpered to select the diskette, depress the **PROGRAM LOAD** switch on the processor.

After MicroNOVA DTOS has been loaded successfully, the following message will be printed on the terminal:

```
TOP OF MEMORY = (e.g. 77777)
DTOS REV xx
```

An asterisk - * - will be printed on the terminal, prompting the operator to begin issuing commands.

NOTE *The primary device code for the diskette subsystem is 33_8. The device code used with a second, third, or fourth diskette subsystem is 73_8, 30_8, or 70_8 respectively.*

RESTART PROCEDURE

If the processor halts while a diagnostic is running, the monitor may be reloaded into memory by restarting at MEMTOP.

This page intentionally left blank.

APPENDIX B
PROBLEM ISOLATION

INTRODUCTION

An operator should have a copy of the program listing for each diagnostic that will be loaded. Appendix G contains listing numbers for each diagnostic that runs under microNOVA DTOS.

MicroNOVA DTOS can be used in a variety of ways. However, to isolate a problem when it occurs, it is best to use a systematic approach. Such an approach is described below.

DAILY ROUTINE

The operator should issue the QUIT command at the close of each day. This command loads an exerciser program and allows it to run overnight; thus the user can detect any possible problems in the morning, before the system will be used.

IF AN ERROR OCCURS

Under microNOVA DTOS, messages typed to the output terminal inform the operator that an error has occurred. Two types of error messages may be printed: those resulting from errors which occur when the operating system has control and those resulting from errors which occur when a diagnostic is executing. Error messages resulting from a problem when microNOVA DTOS is in control are contained in Appendix E. All other error messages are printed to the terminal by the diagnostics. Diagnostic error messages are contained in the front of each program listing.

When an error occurs the operator may choose either to call Data General field service or to diagnose the problem by referring to the listings and proceed from there. If DGC field service is called, steps 1 through 7 in the procedure described below should be followed before the representative arrives.

RECOMMENDED PROCEDURES

1. Enter the **ACCEPT** command. This will load and execute all diagnostics applicable to the equipment table with the pass counters set to one for each test.

2. If no errors occur, then issue either the **RUN** or **RUNALL** command. These commands load and execute all diagnostics applicable to the equipment table and allow each diagnostic to set the number of passes for each test.

3. If no errors occur, issue the **CRUN** command. This command will test the data channel with each applicable diagnostic.

4. Issue the **LOAD** command to run any relevant peripheral diagnostics.

5. Any failing tests should then be loaded individually with the **LOAD** command (if not already done).

6. A record should be kept at the time of failure. This record should contain the contents of the program counter and the accumulators in addition to any error printout from the diagnostic program. This record should be passed on to the DGC field service representative.

7. Any failing tests may then be loaded individually with the **DEBUG** or the **ODT** command. These commands load the debugger program with the failing diagnostic and allow the operator to use breakpoints to pinpoint the exact location of the failure.

SAMPLE TEST RUN

CRUNALL Command *(with no errors found)*

```
*CRUNALL

DTOS  DISK IS CAT/KITTEN DEVICE

DEVICES FOUND BY THE SYSTEM
NAME          CODE

IOBD          0
TTI           10
TTO           11
DKT0          33

LOAD:
MNLGCT   REV. 00

PASS S
PASS P
PASS
PASS
LOAD:
ARITH TST REV. 00
S
 PASS
 PASS

LOAD:
MOSMEM   REV. 00

000002 - 027777 IS MEM TO TEST
   1 PASS SP
   2 PASS
LOAD:
MNSCMT   REV.00
MOM SC-MEMORY TEST REV 00
TOTAL # 1K'S = 12
PROGRAM RUN LIST
PRG#          NAME
0             DATA=ADDR
1             ISZ-TEST
2             MARCH
3             MASEST
```

```
PASS 0 S
PASS 1
PASS 2
PASS 3

LOAD:
MNIOT     REV. 00
END OF PASS 1

LOAD:
MNMRT     REV. 00

MNMRT REV. 00 09/03/76
TOTAL # 1K'S = 12
PROGRAM RUN LIST
PRG#          DESCRIPTION
0             CHKRBRD RAN.
1             SC MEMORY TEST
2             ARITHMETIC TEST
3             MUL/DIV TEST
4             REAL TIME CLOCK
5             TTY TEST
END RUN  1      -RUNALL MODE

PASS S
PASS P
PASS
PASS S
PASS
```

SAMPLE TEST RUN

SELECT Command *(with errors detected)*

```
*SELECT MNIOT

LOAD:
MNIOT    REV. 00
ERROR NUMBER 1   ENCOUNTERED SUBTEST 1
CRY    AC0    AC1    AC2    AC3    PC
1      000000 177777 000000 000000 000612

DDOS   REV NEW = 0
```

This page intentionally left blank.

APPENDIX C
COMMAND SUMMARY

This appendix contains a brief functional description of each command. It does not include command format and specific information regarding command execution. For a complete and detailed description of each command, refer to chapter III.

ACCEPT	Load and run all diagnostics applicable to the equipment table one time.
ADD	Add the specified device to the equipment table.
ASSIGN	Clear the equipment table and add the specified devices.
DEBUG	Load the specified diagnostic and the debugger, start the diagnostic with debugger commands.
ODT	Load ODT debugger and specified diagnostic; start in ODT
DELETE	Remove the specified device from the equipment table.
DIR	Print program directory on the terminal.
ENTER	Print specified heading on the terminal.
EQUIP	Print current equipment table on terminal.
HELP	Print the command summary and the list of devices auto tested if found on the I/O bus.
LOAD	Load and run the specified diagnostic.
LOAD.N	Load and run the specified diagnostic N times.
QUIT	Load and start the exerciser (MNMRT).
RUN	Load and run all diagnotics applicable to the equipment table one time; internal pass counters set by the diagnostic programs.
RUNALL	Load and run all diagnostics applicable to the equipment table.
SELECT	Load and run the specified diagnostics.
REPEAT	Load and run the specified diagnostics repeatedly.
SWREG	Allow input from the terminal keys for switch settings.

NOTE *To run the data channel test with the diagnostic, preface the LOAD, RUN, REPEAT, RUNALL, and SELECT commands with* **C.**

This page intentionally left blank.

APPENDIX D
EQUIPMENT TABLE

The following devices will be entered into the equipment table automatically by microNOVA DTOS when attached to the I/O bus. This list is printed on the terminal whenever the command, **HELP** is given. To alter this list, use commands described in chapter III. entitled, *Commands Which Modify the Equipment Table.*

DEVICE	MNEMONIC	DEVICE CODE	MODEL NUMBER
I/O Test Board	IOBD	00	-
Hand-Held Console	HHC	04	8564
PROM Programmer	PROG	05	8574
Teletypewriter	TTI	10	-
	TTO	11	-
Second Teletype	TTI1	50	-
	TTO1	51	-
3rd Diskette Subsystem	DKT2	30	6038, 6039
1st Diskette Subsystem	DKT0	33	6038, 6039
4th Diskette Subsystem	DKT3	70	6038, 6039
2nd Diskette Subsystem	DKT1	73	6038, 6039

This page intentionally left blank.

APPENDIX E
ERROR MESSAGES

The following error messages occur during Data Channel excercising:

DURING CAT PROGRAM EXECUTION

(programs: MNCAT,MSCAT)

MESSAGE	REASON
DKP ERROR STATUS = XXXXXX	XXXXXX is a 16-bit error status word indicating the reason for the failure. Refer to the Programmers Reference, Peripherals (DGC # 015-000021).
DISC TIMEOUT	If this message occurs before MNCAT prints an S, it indicates that the diskette took too long to step out a track. If it occurs after the S is printed, the indication is that the diskette could not find the correct sector for its transfer.
I/O ADRS ERR XXXXXX YYYYYY	XXXXXX= correct address YYYYYY= wrong address This message indicates that the diskette made the wrong number of data transfers.
GOOD BAD XXXXXX YYYYYY ADDRESS ZZZZZZ	This message indicates that data channel problems exist: GOOD is the expected data, BAD is the data received from diskette, ADDRESS is the data buffer location where the error occurred.

In the stand-alone version also:

MESSAGE	REASON
INSUFFICIENT MEMORY TO RUN MSCAT	The system has less than 8K of memory.
ERROR IN MOVING PROGRAM: CORRECT WRONG XXXXXX YYYYYY TO ADR FRM ADR ZZZZZZ AAAAAA	The contents of the moved program location (ZZZZZZ) do not match those of the corresponding location (AAAAAA) of the original program.

DURING KITTEN PROGRAM EXECUTION

(program: MKITTEN,MSKITTEN)

MESSAGE	REASON
I/O ADRES ERR XXXXXX YYYYYY	XXXXXX=good address YYYYYY=bad address The I/O test board made the wrong number of transfers.
GOOD BAD XXXXXX YYYYYY ADDRESS ZZZZZZ	Data Channel problems exist. Good is the expected data. Bad is the data received from the I/O test board. Address is the address in the data buffer where the error occurred.

DURING DISC BOOTSTRAP

(program: MBOOT.SR)

The diskette bootstrap program will halt at locations 00073_8 or 77_8.

If the label, DDOS, is missing from the diskette, the diskette or cartridge being used may not be the correct one.

Error halt (plus 1) locations indicating diskette errors are 133_8, 233_8, and 332_8. One of these locations will appear as shown during an ODT typeout, or on the hand-held console display.

DURING DTOS EXECUTION

(program: MMINI.SR)

The mini-monitor will halt at location, $X7677_8$ or $X7766_8$ where X is equal to the number of the highest 4K memory module in the system.

(program: MMON.SR)

MESSAGE	REASON
DISK ERROR	A disc read, or step in/out error has occurred. The operator should run the disc diagnostic or reliability test.
NO SUCH PROGRAM:XXXXX	The specified program cannot be located on the disc.
SYSTEM CONFLICT, CANNOT FIND PROGRAM	The program cannot be located on the disc.
INPUT OVERFLOW	The terminal's input buffer cannot handle the number of characters which have been entered.
NOT ENOUGH MEMORY FOR THIS LOAD	The specified diagnostic requires more space than is available in memory.
ILLEGAL NAME,XXX	The specified name is not correct.
DEVICE CODE NUMBER ERROR,XX	The specified device code is not correct.
SYSTEM CRASH	The system is not functioning, perform a bootstrap procedure as described in Appendix A.

APPENDIX F
USE OF THE SWITCH REGISTERS IN DIAGNOSTIC TESTING

TYPES OF SWITCH REGISTERS

There are two types of switch registers in DTOS that the user can access:

- DTOS switch register
- Diagnostic-program switch register

The DTOS switch register is the switch register for the operating system. When DTOS is first loaded, this register has the default value of 000000. The user may make changes to this register any time that DTOS prompts with an asterisk (*) displayed on the terminal. Then, DTOS maintains this value for the DTOS switch register no matter how many times the different programs are loaded or reloaded.

The diagnostic-program switch register is a register for the currently loaded program. DTOS loads the program switch register with the value of the DTOS switch register whenever either a new program is loaded or the same program is reloaded. The user can then change the switch settings as desired by complementing the switches. However, DTOS maintains the current value of this register only while the program is loaded and running. (Refer to the following section titled "DIAGNOSTIC-PROGRAM SWITCH REGISTER" for information on how to complement the switches.)

SWITCH DESCRIPTIONS

The following list summarizes the switches in the DTOS switch register. The descriptions for switches 0 thru 11 are the same for both the DTOS and the diagnostic-program switch registers. However, the list does not give any descriptions for switches 12 thru 15 because DTOS does not use them. Nevertheless, an individual diagnostic program may use switches 12 thru 15 in its own switch register. (Refer to the text file of each program for information on how the specific program uses these switches.)

SWITCH NUMBER	DEFINITION
Switch 0 = 0	All switches in the register are in the default condition.
Switch 0 = 1	Switches in the register are not in the default condition.
Switch 1 = 0	(Default condition) Loop on error.
Switch 1 = 1	Do not loop on error.
Switch 2 = 0	(Default condition) Print-out to console.
Switch 2 = 1	Do not print-out to console.
Switch 3 = 0	(Default condition) Do not print the percent of failure.
Switch 3 = 1	Print the percent of failure.
Switch 4 = 0	(Default condition) Print the pass count.
Switch 4 = 1	Do not print the pass count.
Switch 5 = 0	(Default condition) Do not print-out to the line printer.
Switch 5 = 1	Print-out to the line printer.
Switch 6 = 0	(Default condition) Do not halt on error.
Switch 6 = 1	Halt on error.
Switch 7 = 0	(Default condition) Do not print summary and/or passing of each subtest.
Switch 7 = 1	Print summary and/or passing of each subtest.
Switch 8 = 0	(Default condition) Print only the first error.
Switch 8 = 1	Print every error.
Switches 9-11	Reserved for future use by DTOS.
Switches 12-15	Not used by DTOS. Used only by the individual diagnostic programs.

DIAGNOSTIC-PROGRAM SWITCH REGISTER

Introduction

The user can change the switch settings in the program switch register during both of the following modes:

- Program-Run Mode
- Switch-Modification Mode

In both of these modes the user complements a switch by pressing a key that corresponds to the switch. This is the only way to change a switch in the program switch register because the "SWREG" command does not work while the diagnostic program is loaded and running.

Whenever the user wants to display the switch register value without changing the switches, he inputs the following command.

COMMAND	RESULT
M	This command can be input at any time to display the current value of the switch register spread out by bits. For example, if the value is an octal 000000, the terminal displays it as: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

The following section titled "Complementing The Switches" describes the input process for both the Program-Run Mode and the Switch-Modification Mode.

Complementing The Switches

The Program-Run Mode

The user complements a switch in the diagnostic-program switch register while the program is running by pressing one of the keys listed below in the section titled "Keys Used To Complement Switches". The program accepts this input as a switch-complement input as long as it is not waiting for another input, such as when the following prompt is output to the terminal: "ENTER THE DEVICE CODE". The terminal then echoes the key that was pressed to verify that the related switch has been complemented.

If the user wants to change more than one switch, he just presses all the related keys with no spaces or commas between them. The terminal then echoes all the keys in the order that they were entered. If the user presses a key that is not listed below, the terminal just echoes the character without changing any switches.

It is important to note that in the Program-Run Mode, the program continues to run after complementing a switch.

Switch-Modification Mode

The Switch-Modification Mode allows the user to stop the normal flow of the diagnostic program while making changes to the program switch register. The user stops the program by inputting the CONTROL 0 command which is explained in the following list. He then complements switches in exactly the same manner that he does in the Program-Run Mode. However, the only difference is that the program is stopped during and after the changes are being made. The user starts the program again by inputting one of the commands listed below.

COMMAND	RESULT
0	This command locks the program in a mode called the Switch-Modification Mode. The user can input this command any time the program is not requesting a specific input (like "ENTER THE DEVICE CODE"). Refer to the above section titled "Program-Run Mode" for details on how to complement the switches by pressing the terminal keys.
Carriage Return	The user usually presses the Carriage Return key after he inputs a change. Using this key does the following: it keeps the switch register set to the most current value, exits the Switch-Modification Mode, and continues with the program from where it was when this mode was entered. If the user presses this key without inputting any changes, the program exits the Switch-Modification Mode, keeps the switch register value the same as it was on entering this mode, and continues with the program from where it left off.
CONTROL R	This is a system control character which normally keeps the switch register set to the current value and restarts the program from the beginning. However, when it is used during the Switch-Modification Mode, it causes the program to exit from this mode in addition to performing the above mentioned functions.
CONTROL D	This is a system control character which normally resets the switch register to the default value (all zeroes) and restarts the program from the beginning. However, when it is used during the Switch-Modification Mode, it causes the program to exit from this mode in addition to performing the above mentioned functions.

The following example shows how the user would change some switches in the switch register.

Example:

KEYS PRESSED	RESULTS
0	The program is now in the Switch-Modification Mode.
M	The terminal displays the current switch-register value spread out by bits. For example, if the value is an octal 000000, the terminal displays it as follows: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
46A	Switches 4, 6, and 10 are complemented. The switch-register in the above example is now octal 105040. (See note below.)
M	The "M" command is entered again just to see the changed bits as follows: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 1 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0
Carriage Return	The program exits the Switch-Modification Mode; the switch register has the new value of 105040; and the program continues where it left off.

NOTE *The program sets bit 0 to a value of 1 whenever another bit in the switch register is set, otherwise it resets the bit. This explains why bit 0 in the above example (105040) is a 1 even though it was not set by pressing key 0.*

Keys Used To Complement Switches

The following list shows the keys that are used to complement each switch.

KEY	SWITCH
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15

This page intentionally left blank

APPENDIX G
DIAGNOSTIC ABSTRACTS

This appendix contains the program name, binary tape number, listing number, program mnemonic, and abstract for each diagnostic used with microNOVA DTOS.

The diagnostics are listed alphabetically by mnemonic.

ARITHMETIC TEST

Binary Tape: 095-000444
Listing: 096-000444
Mnemonic: ARITH TST
Abstract: This program is designed to exercise the arithmetic and logic instructions. The program adjusts its parameters to the size of memory and relocates itself to various areas of memory. It should be run to ensure proper operation of the memory system.

DISKETTE DIAGNOSTIC

Binary Tape: 095-000445
Listing: 096-000445
Mnemonic: FPY DIAG
Abstract: This program is used to perfom comprehensive testing on the diskette controller.

DISKETTE FORMATTER

Binary Tape: 095-000443
Listing: 096-000443
Mnemonic: FPY FRMT
Abstract: This program is used to write the correct track address into the preamble block which prefaces the data of each sector, within the domain of any specified track. Formatting is limited to one drive at a time.

DISKETTE RELIABILITY

Binary Tape: 095-000442
Listing: 096-000442
Mnemonic: FPY RELI
Abstract: This is a maintenance program designed to exercise and test the diskette controller and 1-2 diskette drives.

ASYNCHRONOUS INTERFACE

Binary Tape: 095-000446
Listing: 096-000446
Mnemonic: MNASYN
Abstract: This is a program which verifies the proper operation of the asynchronous interface, including the ODT option, if present.

HAND HELD CONSOLE TEST

Binary Tape: 095-000447
Listing: 096-000447
Mnemonic: MNHHC
Abstract: This program tests the function keys, the display and the logic circuit.

microNOVA I/O TESTER

Binary Tape: 095-000448
Listing: 096-000448
Mnemonic: NMIOT
Abstract: This program is used in conjunction with a test board to aid in the checkout of the I/O bus, interrupt and data channel. The test board eliminates the need to have a device which uses all of the processor I/O features.

microNOVA LOGIC TEST

Binary Tape: 095-000423
Listing: 096-000423
Mnemonic: MNLGCT
Abstract: This is a maintenance program designed to test the central processing unit. It is a gate-by-gate test of the logic used to implement the instruction set. Read back of the pattern is proof that all locations exist. The I/O equipment is not tested.

LINE PRINTER TEST

Binary Tape: 095-000537
Listing: 096-000537
Mnemonic: MNLPT-TST
Abstract: Line printer test 1 is a maintenence program designed to test line printer types 4034A (Data Products), 4034B (Data Products), 4034C (Centronics), 4034D (Centronics). This program will also test the zone select and format control options.

MICRONOVA MULTIPROGRAMMING RELIABILITY TEST (SHORT VERSION)

Binary Tape: 095-000424
Listing: 096-000424
Mnemonic: MNMORT S
Abstract: This test consists of a series of individual processor and peripheral tests and a supervisor program, the diagnostic linker. The linker is a program designed to "link" the C.P.U., memory, multiply/divide, TTY, and real time clock tests.

MICRONOVA MULTIPROGRAMMING RELIABILITY TEST - LONG VERSION

Binary Tape: 095-000506
Listing: 096-000506
Mnemonic: MNMORT L
Abstract: This test consists of a series of individual processor tests, peripheral tests and a supervisor program, the diagnostic linker. The long version includes all of the short version tests and primary and secondary device code tests for the 6038/39 disk.

microNOVA READER, PUNCH, CLOCK DIAGNOSTIC

Binary Tape: 095-000538
Listing: 096-000538
Mnemonic: MNRPC
Abstract: The reader/punch/clock diagnostic is designed to fully test the microNOVA paper tape reader/real time clock option board, and to provide an operational test of the paper tape reader and/or punch peripherals.

microNOVA POWER SHUT DOWN TEST

Binary Tape: 095-000441
Listing: 096-000441
Mnemonic: MNPWR
Abstract: This is a maintenance program designed to test the power monitor and autorestart. The program also tests for memory retention upon power shut down and verifies the AC contents.

microNOVA SC MEMORY TEST

Binary Tape: 095-000422
Listing: 096-000422
Mnemonic: MNSCMT
Abstract: This test consists of a series of SC memory tests and a simple supervisory program, the diagnositc linker. The diagnostic linker is a program designed to "link" the variety of SC memory tests.

MOS MEMORY TEST

Binary Tape: 095-000440
Listing: 096-000440
Mnemonic: MOSMEM
Abstract: This is a program which runs a full sequence of MOS Memory tests on the entire memory.

microNOVA PROM BOARD PROGRAMMER DIAGNOSTIC

Binary Tape: 095-000535
Listing: 096-000535
Mnemonic: PBP DIAG
Abstract: This is a maintenance program designed to test the prom board programmer model 8574. The program exercises all relays and programming functions without actually programming and proms.

PRINTER ACTION TEST

Binary Tape: 095-000437
Listing: 096-000437
Mnemonic: PRINT-TST
Abstract: This program provides diagnostic and reliability exercisers for a printer. The printer may be on a TTY (Device Code 10-11, 50-51), or on a QUAD-MUX interface. The printer may be a line printer or data-channel line printer.

APPENDIX H
SYSTEM PROGRAMS

Following are the listing numbers for DTOS system programs:

PROGRAM NAME	LISTING
MNDTOS DATA CHANNEL EXERCISER WITH DISKETTE (MNCAT)	096-000462
MNDTOS DATA CHANNEL EXERCISER WITH I/O TESTER (MKITTEN)	096-000463
STAND-ALONE DATA CHANNEL EXERCISER W/DISKETTE (MSCAT)	096-000464
STAND-ALONE DATA CHANNEL EXERCISER W/I/O TESTER (MSKITTEN)	096-000465
MNDTOS ODT	096-000466
DEBUGGER	096-000372
HELP PROGRAM-COMMAND SUMMARY	096-000373
MNDTOS BOOTSTRAP	096-000459
MNDTOS MONITOR	096-000457
MNDTOS MINI-MONITOR	096-000458

This page intentionally left blank.

APPENDIX I
DIAGNOSTIC OPERATING INFORMATION

The following procedures are designed to outline the operator interface required for running the microNOVA diagnostics. The diagnostics may be run under the DTOS operating system, or using the Stand-alone system configuration. The microNOVA DTOS system requires a diskette subsystem which contains the supervisory programs and diagnostic tests.

Every diagnostic is subdivided into small tests. Starting with the least complicated section of the hardware, each test checks out a part of the logic. As the program flow progresses, sections that have been tested are assumed functional and errors related to them may not be repeated.

It is recommended that MNLGCT, MOSMEM and ARITHTST be run in order to establish the integrity of the processor before running other diagnostics.

MACHINE REQUIREMENTS

microNOVA Processor
4K read/write memory (MNMORT requires 8K)
Teletypewriter
Diskette
Hand Held Console, Console Debug, or
Program Load Option

STARTUP PROCEDURE

Load the program via the binary loader and start at location 200_8. See the individual programs for restarting procedures. The programs will run until manually stopped or the control is returned to DTOS. At the end of each program iteration, completion of the pass will be indicated.

SWITCH SETTINGS

A software location called SWREG is used by the diagnostics that can run under different modes. The address of this location can be calculated by adding 5 to the contents of location 45_8. Location SWREG is used to select the program options (not system configuration). While running under DTOS, this location will be loaded by the Monitor. However, under Stand-alone and Program Load modes, this location will be set according to the parameters supplied by the operator. The options can be changed or verified by using one of the control commands given in Appendix F.

Switch Options

The different bits and their interpretation at location SWREG are covered in Appendix F.

Switch Commands

Diagnostics equipped with the SWREG option offer the following features. Once the program starts executing, the state of any of the bits can be changed by hitting keys 1 through 7. The program will continue running after updating the options. Each key will complement the state of the bit affiliated with it; thus, bit 4 can be altered by hitting key 4. Setting any bit in location SWREG will set bit 0. Default mode is defined as all bits of SWREG set to 0. The program can be locked into switch modification mode by typing a 0, in which case more than one bit can be changed before the control is allowed to return to the main program.

OCTAL DEBUG TOOL (ODT)

After setting the parameters, diagnostics equipped with a built-in ODT can be accessed at any time during execution of the program by hitting Control O. On entering ODT, the address of the location having the next instruction to be executed will be typed out.

Conventions and Symbols

The following conventions are used by the ODT:

? An illegal key has been pressed.

! @ ODT is ready and at your service.

Command Structure

An ODT command has the following format: [argument][command]. An argument may be one of the following:

EXP	An octal expression consisting of octal numbers separated by plus (+) or minus (-) signs. Leading zeros need not be typed.
ADR	An address is the same as an expression except that bit 0 is neglected.

A command is a single teletypewriter character.

ODT Commands

The locations that can be examined and modified by the user are called cells. These cells are of two types: internal CPU cells and memory locations.

Opening Internal Cells

The command to open one of the internal registers is of the form "nA" where n is any octal expression between 0 and 7.

0-3	For accumulators 0-3
4	For PC of the next instruction to be executed in the event of a "P" command.
5	CPU and TTO status Bit Interpretation 13 Status of carry bit 14 Status of interrupts 15 Status of TTO DONE flag
6	Address of the location having break point (if any)
7	Instruction at the break point location

Other Commands to Open Cells

ADR/	Open the cell and print its contents.
./	Open the cell currently pointed to and print its contents.
.+ADR/	Add ADR to the pointer, open the cell and print its contents.
.-ADR/	Subtract ADR from the pointer, open the cell and print its contents.
CR	The RETURN key is used to close the open cell with or without modifications.
LF	Line Feed is used to close the open cell with or without modification and to open the succeeding cell.
/	Close the open cell without modification, and open the cell pointed to by its contents.
+ADR/	Close the open cell without modification, and open the cell pointed to by its contents + address.
-ADR/	Close the open cell without modification, and open the cell pointed to by its contents - address.

Modification of a Cell

Once a cell has been opened, its contents can be modified by typing the new value that the cell is to contain in the form of an octal expression followed by "CR" or "LF". If a + or - is typed as the first character of the expression, then the value of the expression is added to or subtracted from the old contents of the cell. The address itself, or an expression relative to the address, can be deposited by typing a "." or ".+/-octal expression".

Further ODT Commands

RUBOUT This key is used to delete erroneously typed digits. Each time this key is pressed the right-most digit is deleted and echoed on the terminal. If the RUBOUT key is hit immediately after opening a cell, the right-most digit of its contents is deleted. This cell can now be modified as if its contents were typed-in by the user just before the RUBOUT key was pressed.

ADRB Insert a break point at location ADR. Only one break point can be inserted and any entry to ODT after executing a break point will cause it to be deleted.

D Delete the break point, if any.

P Restart the execution of the program at the location pointed to by the register, 4A.

ADRR Start executing the program at ADR after an IORST.

K Kill the string typed so far. The ODT responds with a "?" and the open cell is closed without modification.

TYPES OF ERRORS

The programs are designed to encounter the following types of errors.

Fatal Error

This will be an error caused by something other than the unit under test. An example of this type of error is program flow being out of sequence. A fatal error causes the program to halt. The program should not be continued after a fatal error.

Soft Error

This error will be reported when there is a problem with the unit under test. After reporting the error, the program acts according to the settings of the SWREG. The program can be continued after a halt on a soft error.

DESCRIPTION OF INDIVIDUAL DIAGNOSTICS

This section describes the additional operator interface required for running each of the diagnostics along with their error descriptions and exceptions.

MNLGCT

This diagnostic cannot be run under the different optional modes described in Appendix F. Optional starting addresses are 170_8 without CAT/KITTEN and 171_8 with CAT/KITTEN (must have been previously loaded). During the first pass the program halts at location 503_8 to verify that the CPU can execute a halt. The program halts on finding an error.

ARITHTST

Bit 7 of SWREG does not apply. After error printout, refer to the listing for a description of the accumulator contents.

MOSMEM

This diagnostic requires only 1K of good memory and a teletypewriter. Optional starting addresses are 170_8 without CAT/KITTEN and 171_8 with CAT/KITTEN (must be previously loaded). Bit 7 of SWREG, Switch commands and ODT commands do not apply.

User's Options

Location USREG (address = 6+contents of location 45_8) is used to define one of the following options:

TEST 1 Check the external address circuitry.

TEST 2 Check the external data circuitry.

TEST 3 Check the memory for data holding capability.

TEST 4 Check the volatility of the memory cells.

TEST 5 Marching ones and zeros.

TEST 6 Shifting diagonal test.

TEST 7 Check the worst case address and data transitions.

TEST 10 Row/column or long galloping test on sequential cells.

TEST 11 Row/column galloping test on cells, 32 or 64 locations apart.

TEST 12 Read memory through indirect addressing.

BIT(S)	VALUE		INTERPRETATION
	OCTAL	BINARY	
0	—	0	Check the entire memory.
	100000	1	Memory boundaries will be defined by the operator.
1	—	0	Memory board is designed around 1K RAMs or DGC 4K RAMs.
	040000	1	Memory board is designed around other 4K RAMs.
2	—	0	Allow relocation of the program to the highest available memory.
	020000	1	Suppress relocation.
3	—	0	Only row/column galloping test is to be performed.
	010000	1	Enable long galloping test.
1	—	0	Do not loop on a test.
	000020	1	Loop on the test specified by bits 12-15.
12-15	—	—	Test number. Only test defined by bits 12-15 through the last test will be executed.

Soft Error Format

When a bit fails for the first time, an error with the following format is reported:

BDATA - GDATA - LOC - TSTNM - PC

where:

BDATA is the bad or found data.

GDATA is the good or expected data.

LOC is the address of the failing location.

TSTNM is the test number.

PC is the PC where the error was detected.

Error History

On completion of a pass, the error history of the failing bits will be reported unless SWREG is set to suppress the end of pass type out, in which case, the information will be collected until it is obtained by hitting any TTY key. The error history will be for the time since it was reported last.

ERROR HISTORY FORMAT - The error history will be reported in the following format:

BITNM - QUAD - BANK - ERTMS

where:

BITNM is the failing bit number (0-15).

QUAD is the 1K quadrant of the memory bank. Each bank is divided into 4 quadrants. Location 0 will be reported in Q0 of B0.

BANK is the bank number. The memory is divided into 4K banks. Bank 0 is $0\text{-}7777_8$, Bank 1 is $10000\text{-}17777_8$, and so on.

ERTMS is the number of times the bit failed reported in decimal. Highest number reported will be 32767_{10}.

TTY Commands

Any key will cause the error history to be typed out. The program will continue running.

CNTRL D This command given at any time will restart the program and the switch options will be requested.

CNTRL C This command given at any time will cause the program to be written into its original location and halt.

NOTE *A CONTINUE after CNTRL C will start the program with the previous settings of the options. The program can be started at location 200_8 to reset the options.*

MNSCMT

Bits 3 and 7 of SWREG do not apply. ODT commands do not apply. Starting at location 206_8 enables the selection of each test.

Additional keyboard commands are:

CNTRL S Print the run statistics of each test.

Q Use all 1's data pattern in MARCH, GALPAT, and GALWREC tests.

P Print pass for each test completed.

T Allow entry of memory test limits.

Definitions of error printout terms:

C(x) Contents of location x.

C/(x) Complement of the contents of location x.

LOC(x) Address of location x.

SCRLO/HI Logical scratch limits of this pass of the test. (Scratch memory is the memory under test.)

PHSLO/HI Selected test limits in decimal 1 K's.

MNPWR

This diagnostic cannot be run under different optional modes described in the switch setting and ODT sections.

MNASYN

Hardware Connections

Disconnect the terminal plug from the board to be tested and install the special loopback patch plug with the following connections:

Transmit Data to Receive Data	A21-A1
Data Terminal Ready to Clear To Send	A10-A2
Data Terminal Ready to Data Set Ready	A10-A6
Request To Send to Carrier Detect	A12-A4
Request To Send to Ring Indicator	A12-A8

The unit under test must have jumper W27 removed and must be patched for an 8-bit character length (jumpers W18 and W19 removed). Any baud rate may be used, and transmission may be via either an EIA RS-232 line or a 20mA current loop.

Switch Settings

Switch register options are as described under Switch Options in Appendix F, but Switch Commands and the ODT section do not apply. Setting bit 15 allows suppression of calculated baud rate printout. Bit 14 of SWREG causes all references to the UUT to be changed from the default value found in UTAD. Bit 13 performs a similar operation for the system console address using location CONAD.

Error Description

This program is designed to provide a *snapshot* of all register contents when an error is detected. These are saved in a memory stack and, depending on the SWREG options, reported to the user via some listing output device. A running count of errors is saved by the program and facilities are provided for looping or halting on error detection. Memory as follows:

After a halt on error, the contents of the CPU registers at the time of the failure may be found by examining the contents of the stack. The stack pointer, SP, contains the address of the failing PC with bit 0 of that word containing the carry bit. SP-1 contains AC3, SP-2 contains AC2, SP-3 contains AC1 and SP-4 contains AC0. Other information may be obtained from page 0 locations TSN?M which contains the failing subtest number and ERN?M which contains the error number.

Detailed error messages print out the register contents at the time of the failure, as well as identifying the subtest and error number causing the problem. The message format is:

```
error number xxx encountered subtest xx
cry  ac0  ac1  ac2  ac3  pc
x  xxxxxx xxxxxx xxxxxx xxxxxx xxxxxx
```

These register contents should be compared to those desired by the failing subtest to isolate the error.

MNHHC

Wire W1 must be removed from the hand-held console board, and the CPU placed in lock mode before the test is started. Switch register options are as described, except that bit 15 of SWREG is used to allow error reporting on a partially operational HHC and bit 14 to suppress operator prompts. Console error reporting is identical to the section above.

FPYDIA

To loop on a test, a "J" must be typed while the tests are running. After answering all questions, the program will loop on that particular test.

TEST 1	Clear status and test that error bits are not set.
TEST 2	Test head home bit in status.
TEST 3	Test head load bit in six settle times.
TEST 4	Test checkword error and error bit.
TEST 5	Test illegal and error bit.
TEST 6	Test buffer address increments properly for a read and a write function.
TEST 7	Test sector error and error bit.
TEST 10	Test floppy mask bit.
TEST 11	Write, read and compare all patterns.
TEST 12	Test head load after a drive change.
TEST 13	Test head settle time.
TEST 14	Test both drives of a dual floppy.
TEST 15	Test program load.
TEST 16	Test B-register bits.

The Command String Interpreter is entered by typing an "I" while the tests are running. Type the command string according to the print out.

The Function Interpreter is entered by typing a "CNTRL F" while the tests are running. Type function (DOA, DOB, DIA, etc.) as indicated by message.

Typing a "CNTRL S" while tests are running will give a summary of error bits detected while running the diagnostic.

To determine what the contents of the AC's contain for an error printout, refer to listing.

FPYRL

This program does not have the ODT option and bit 7 of SWREG does not apply.

The following responses and results apply to the program's request for a test number:

Typing a carriage return causes tests 1 through 9 (see below) to be run in sequence.

Typing one of the numerals, 1 through 8, selects and locks on to the corresponding test number. (When the program is locked onto one of these tests the units cannot be changed.)

Typing the "ALPHA",I selects the interchangeability test.

Typing any alpha character moves control to the Command String Interpreter.

The following is a list of the tests and the types of data.

TEST 1	All zeros.
TEST 2	All ones.
TEST 3	Alternate ones and zeros.
TEST 4	Alternate zeros and ones.
TEST 5	All sixes.
TEST 6	Seven zero pattern.
TEST 7	Random data.
TEST 8	Randomly organized ones and zeros.
TEST 9	Unit switching test.
TEST 10	Interchangeability (restarts automatically when completed).

Command String Interpreter

The program has a Command String Interpreter which is entered by typing any alpha character when the test number is requested. When entered, it requests the number of the unit to be tested. A zero or one are the only responses allowed. The data type is then requested. The operator may select any one of the types listed above by typing its corresponding number followed by a carriage return. The command string may now be entered, using one of the following commands: WRITE, READ, SEEK, RECAL, RESTART, LOOP, LR (LOOP RETURN). The WRITE command is restricted to writing one sector or all sectors because of hardware constraints. The RESTART command takes you out of the Command String and back to the start of the program. The format for the Command String is:

SEEK 56 WRITE 11 READ 11 RECAL LOOP

NOTE *A SEEK command must be followed by some other command such as READ, WRITE, etc. A SEEK cannot be terminated with a carriage return.*

The LOOP command allows looping on the command string or the portion of the string preceding the loop command. Exiting from the loop is done by hitting a key on the keyboard. When data is requested the response is a number from 1-6. Listed below are the corresponding patterns:

1 = all zeros
2 = all ones
3 = 125252
4 = 52525
5 = 66666
6 = 70707

FPYFR

This program cannot run under different optional modes described.

MNMORTS

This test does not use the standard microNOVA switch package error routines. This version contains tests that apply to the CPU, memory, multiply/divide, TTY and internal real-time clock. CAT/KITTEN may be used if an I/O tester or floppy disk drive is available.

Bits 3 and 8 of SWREG do not apply. Bit 1 of SWREG means: do not release the scratch area on an error (do not relocate). Bit 4 of SWREG means: print run time.

The program has the following 4 entry points:

200	Run with all the tests selected.
202	Allow the selection of each test.
204	Restart the last selected tests.
210	Immediately enter ODT.

ODT in this program does not have an accumulator and an additional command "=" which causes the current argument to be printed.

Additional keyboard commands that can be executed while the program is running are as follows:

"	Do not print memory allocation table.
#	Delete random wait states in the TTY tests.
CNTRL S	Print the run statistics of each test.

Error Description

Most errors detected by either the individual test programs or by the diagnostic linker will result in an extensive error typeout. A small number of highly improbable errors may result in a program halt if they are of a nature that the linker cannot recover and logically proceed (e.g., interrupt stack overflows).

Error Format

Error typeouts include:

Program number and name at time of error.

The current contents of AC0, AC1, AC2.

Logical scratch and data channel limits.

Continuation information in groups of 3.

Memory locations pertinent to the individual test that failed.

In the error typeouts, the CPU tests that relocate will contain:

ST.LA START/ERROR (RES.)
xxxxxx yyyyyy zzzzzz

Where:

ST.LA	Is the logical start of the relocated test loop (i.e. the last LCALL SETUL).
START	Indicates where the resident copy of the test loop may be found in the listing.
ERROR	Indicates where in the resident copy of the listing the error call may be found.

Error Analysis

Due to the interactive nature of the tests involved, a series of error messages will probably be required for analysis before a problem can be isolated. A restart at location 202_8 and a deletion of all but the test that originally failed may help to isolate interactive problems as follows: If the test runs by itself, the problem is interactive. In this case, re-enable one test at a time to determine which one is the problem. If the test does not run by itself, perform similar but lower level tests for isolation.

Pertinent Memory Locations Typed

Checkerboard Ran

The AC's at error will indicate: GOOD DATA, BAD DATA, LOGICAL ADDRESS

In addition, the following locations are typed:

CB.TK	Test counter.
0	Generate checkerboard.
1	Disturb pass.
2	Check pattern.
3	Checksum, the number of -1's in the pattern.
CB.LC	Starting logical address of "begin" relocated to scratch.
CB.SE	AC3 at error call.

microNOVA SC Memory Test

This is an ISZ/DSZ test for SC-memories. The AC's at error will indicate:
ACTUAL, EXPECTED, LOGICAL ADDRESS

In addition, the following locations are typed:

MM.TK	Error number:
0	Pattern storing error (should be -1)
1	Location not -1 before doing ISZ
2	ISZ didn't skip
3	Location not equal to 0 after ISZ
4	DSZ skip error
5	DSZ test, location not -1 after DSZ
6	Same as 1, except testing in rev direction
7	Same as 2, except testing in rev direction
10	Same as 3, except testing in rev direction
MM.SE	Instruction address following error call

Arithmetic Test

The AC's will be typed as they were at the time of error detection.

In addition, the following locations are typed:

AT.LC	Starting address of ARITH in scratch.
AT.LO	Low limit of scratch area after it is remapped for execution.
AT.LA	AT.LC in relation to AT.LO (logical start of arith after remapping).

The last three random numbers generated.

MUL/DIV Test

Multiply/Divide failures will indicate either MUL for multiply or DIV for divide. In addition, three sets of AC's are typed:

Original operands

Hardware result (assumed to be incorrect)

Software result (assumed to be correct)

Power Fail Interrupt

Upon detection of a power fail interrupt the logical address of the PC at interrupt will be saved. If auto-restart is enabled or the power fail was only momentary, the test will restart as in a start at location 206_8 after typing power fail xxxxxx (where xxxxxx is the PC at interrupt).

Illegal Supervisor Call

Upon detection of a supervisor call, which did not match the list of subroutine calls, the following message will be typed:

```
ILLEGAL SUPER CALL AT xxxxxx

PROG# nnn

AC's qqqqqq yyyyyy zzzzzz

tttttt wwwwww ssssss
```

Where xxxxxx is the logical address of the SUPER CALL, tttttt is AC3 contents and wwwwww is the physical page number, and ssssss is the instruction causing the SUPER CALL.

Interrupt Wait Elapsed

The peripheral device associated with the program number typed has not responded with a program interrupt for an extended period of time. The second number typed should point to the interrupt handler for the device that failed.

MORT L

This test includes all tests of MORT S as well as tests of the 6038/39 disk system and the microNOVA I/O tester. This version cannot be run with CAT/KITTEN. Additional errors are described below:

Error Formats for Disk

I. Data Errors:

A. AC0 contains BAD DATA
B. AC1 contains GOOD DATA
C. AC2 contains ADDRESS OF BAD DATA
D. AC3 contains ADDRESS OF ERROR

II. Status Error AC0-2 contains the status word.

III. Invalid Track Address:

A. AC0 contains ACTUAL TRACK ADDRESS
B. AC1 contains MAXIMUM LEGAL ADDRESS

IV. Disk Seek Errors:

A. ACO contains ACTUAL TRACK ADDRESS
B. AC1 contains EXPECTED TRACK ADDRESS

In other errors the accumulators are not used. However, the following information is also typed:

FYDST	Start of write data in memory. Read data starts at this location +256.
LAST DOB	Address of last transfer.
STATUS	Last disk status.
LAST CMD	Last command to disk.
RETRY □	Number of retries attempted.
BLK □	Data block in error.
UNIT	Unit no. under test.
TRACK	Track no. under test.
SECTOR	Sector no. under test.

Two conditions can cause the program to halt. They are:

I) Drive selection error - drive no. in status word not same as drive selected.

II) Expected interrupt did not occur.

Error Formats for I/O Tester

Accumulator contents for a write sequence - verify buffers, and for a read sequence are in the same format as data errors on the disk. Accumulators for a write sequence - verify address are:

AC0 contains BAD ADDRESS
AC1 contains GOOD ADDRESS
AC2 contains ADDRESS OF BAD ADDRESS
AC3 contains ADDRESS OF ERROR

Additional information provided includes:

W/R	Write/read flag. 0=write to tester, 1=read from tester.
IO.BK	Number of 64 word blocks.
B#ERR	Block containing error.
IODST	Start of buffer area.
IODIB	Address of last word transferred.
ERRDST	Address of bad data word.

MNIOT

Equipment required: microNOVA with I/O tester.

This test will confirm the proper operation of the I/O tester if errors occur when programs are run with CAT/KITTEN.

This is used by other diagnostics as an exerciser for the memory bus when they are run with CAT/KITTEN, so the diagnostic should be run if failures occur during CAT/KITTEN operation.

MNRPC

The device codes for each device to be tested must be entered on the console when requested. The reader portion of the test requires a test tape which is created by the punch portion of the test. This tape must be placed in the reader with a few inches of leader preceding the data. Special looping on data during the punch and reader tests is allowed using switch register bits 14 and 15. Bit 15 causes looping on all data patterns, and bit 15 causes looping on the current data pattern. Other switch options and error reporting are as described in the program listing for error interpretation.

PBP DIAG

Equipment required:

microNOVA with 4K read/write mem
Teletype
Prom board programmer

The PBP must be plugged into the main chassis frame and not connected to a prom board for these tests.

Program description:

The program is in three parts:

1. A diagnostic test to determine if certain logic elements are properly performing.
2. A scope loop to see if the voltages and timing relationships are correct.
3. Another scope loop to check the addresses.

Starting procedure:

Load the program and start at 200, the program will ask:

Select
(1) diagnostic
(2) timing loop
(3) address loop?

A response of either 1, 2 or 3 followed by a carriage return will select the desired test.

Any selected test will run continuously.

If the ESC key is struck the program will stop and give another select message.

MNLPT-TST

The word pass will be printed on the teletype at the end of each iteration of the diagnostic.

When an error is detected by the diagnostic, the program will halt at the location of the error +1. The contents of accumulator 3 will be the address of the failing routine. (Consult the listing to determine the cause of the error.) Press continue to set-up a failing loop suitable for scoping.

Use the following switch settings after a halt due to an error:

A. switch 3 (1) print the failure rate.
B. switch 1 (1) proceed to the next test.

A special carriage positioner test for printer types 4034C and D is provided. The test provides for visual verification of proper operation. The first section of the test prints a line of letters diagonally over several pages, from column 1 to N. The second section prints a solid right triangle, formed by position printing of one letter.

When a halt is executed in routines other than the diagnostic, consult the listing for the cause of the error.

Starting addresses:

200 = Execute all tests
501 = Diagnostic
502 = Print "E"
503 = Print "="
504 = Print "M"
505 = One letter each line
506 = Print character set
507 = Zone print option
510 = Format control option
511 = Keyboard echo on printer

This page intentionally left blank.

READERS COMMENT FORM

DOCUMENT TITLE: .....

Your comments, accompanied by answers to the following questions, help us improve the quality and usefulness of our publications. If your answer to a question is "no" or requires qualification, please explain.

How did you use this publication?

( ) As an introduction to the subject.
( ) As an aid for advanced knowledge.
( ) For information about operating procedures.
( ) To instruct in a class.
( ) As a student in a class.
( ) As a reference manual.
( ) Other.....

Did you find the material:

- Useful.....YES ( ) NO ( )
- Complete.....YES ( ) NO ( )
- Accurate..... YES ( ) NO ( )
- Well organized..... YES ( ) NO ( )
- Well written.....YES ( ) NO ( )
- Well illustrated.....YES ( ) NO ( )
- Well indexed.....YES ( ) NO ( )
- Easy to read..... YES ( ) NO ( )
- Easy to understand..... YES ( ) NO ( )

We would appreciate any other comments; please label each comment as an addition, deletion, change, or error and reference page numbers where applicable.

CUT ALONG DOTTED LINE

COMMENTS

PAGE	COL	PARA	LINE	FROM	TO

From:

NAME..... TITLE.....
FIRM..... DIV. .....
ADDRESS.....
CITY..... STATE..... ZIP.....
TELEPHONE..... DATE.....

Data General Corporation

ENGINEERING PUBLICATIONS
COMMENT FORM
DG-00935

FOLD DOWN FIRST FOLD DOWN

FIRST CLASS
PERMIT NO. 26
SOUTHBORO
MASS. 01772

NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

Postage will be paid by:

DataGeneral
Southboro, Massachusetts 01772

ATTENTION: Engineering Publications

FOLD UP SECOND FOLD UP

STAPLE